# BuBoard

## Project Requirements and Specification

October 6th, 2017

Prepared For Ethan Murphy and Karan Shrestha,
Under the Direction of Dr. Clifton Presser and Dr. Rodney Tosten, Gettysburg College

Prepared By
Michael Clark-Cuadrado
Zachary Miller
Jingyi “Jannie” Liu

*{clarmi03, millza04, liuji02}@gettysburg.edu*

# Introduction

At Gettysburg College, our clients have observed a lack of accessible methods of communication within the student body. One method through which the student body can communicate with itself is through a college-sanctioned and operated daily newsletter called "The Digest". This newsletter is open for anyone to add an announcement, effectively reducing its meaningfulness. This results in low readership and participation amongst students, which causes the information put into it to be conveyed in an ineffective manner. The solution presented to our team, conceived by our clients, is to develop a web based application that would display the information in a more effective manner. The expectation here is to emulate the methods of social media sites that have made them an useful source of timely information. Our exercise for this semester is to develop this application, for the purposes of allowing our clients to perform user testing, validation, and viability tests on their idea.

This document will describe the application as we understand its requested application specification to be; it does not describe what we believe is technically feasible, or what we believe is reasonable to develop in a given amount of time, but exists to facilitate a common understanding of the project as it was given to us. We anticipate this document will be amended and adjusted as we become more intimately acquainted with the minutiae of the project during development.

## Scope

The application we are developing is intended to be used in academic departments of Gettysburg College, in order to test usage patterns. Certain departments would use the

application and give feedback on what can be done in the future to improve. After the design is seen to be in a working state, it will then be deployed as part of a venture by our clients. After this test phase, it is hoped that other colleges will adopt this technology and integrate it into their schools. The final plan for this website is to have a website which can be deployed across multiple colleges, each having their own local board.

### Purpose

The main purpose of this project is to increase inter-campus connectivity. It has been observed that many on campus have trouble communicating with one another through college-sanctioned means. The only way currently is through the Gettysburg Digest. This Digest has become bloated and unappealing to users over the years. Where so much information is contained in it, and so little is relevant to the individual student. With this new system, it is hoped that it will allow students to filter through any useless information easily and effectively to find information they care about. The two main features here will be for students to be able to post and for students to be able to follow others who post. This could be a student following a teacher who is posting assignments, or a club member following his/her club's related activities. The hope is that this will filter through the useless information, giving students easy access to relevant information.

## Requirements

### Format

The given specification dictated that the application should take the form of a web-based application, accessible from web browsers. An item of particular importance is the ability to be

accessible from a mobile device as well as from desktop computers. In order to leverage our group's existing technical abilities and experience, and in order to make this web-based application, we chose standard technologies that we will use during development.

For the backend of the application, which is executed on the server, will be using the PHP language, and the MySQL database system for storing our application's data. For the front end of the application, which is executed in the user's browser, we have chosen a design library called Material Design Lite, made available from Google, which will ease development of polish and experience, as well as a javascript framework named jQuery, to avoid wasting effort on boilerplate components and code. We are also considering testing the viability of a few other frameworks to work in conjunction with those previously stated, namely Vue.js, React, and AngularJS.

**Functions Performed**

The application centers around the creation and sharing of posts, and the creation and 'following' of user accounts. The main functionality is the ability to view the posts made to your local group, sort and search them. In order to facilitate this, it is possible to create a list of subscribed authors, who posts are accessible in one single feed, known as the personal feed.

For example, if you were a student or campus figure, and you wanted to see the announcements or events that were made recently, you would access the BuBoard site, log into your account, and view the general board. You could narrow down the board to only announcements or only events, or whichever tag you are interested in. If you had a certain group

of friends that you were interested in following, you could add each of those friends to a follow list, and get a curated list of posts based on their topics.

## Technical Structure of Application

This section will detail a preliminary design of the application that we believe meets the stated requirements of the solution. The design diagram is broken down into three sections, modelled after the Model-View-Controller format. In this scheme, all parts of the application can be divided into either Models, Views, or Controllers. **Models** describe the different kinds of data objects that the application will store, retrieve, modify, and process during its operation. **Views** are the different screens or pages that the user may encounter, and are described by components visible on them, and actions that the user may take. **Controllers** are the snippets of code that perform a function, which typically changes a view, modifies a data object, or performs some other function.

**Views**

Our clients have provided several wireframes for the mobile version. However, they wanted this program to be a web application first. We try to maintain the consistency across different platforms, e.g. mobile, tablet, computer, etc. Thus, the user interface for mobile and computer will share large similarities, but with small variations in structures and positions. I will include both the mobile wireframes that they provided and the wireframe that we designed for web application in this section.

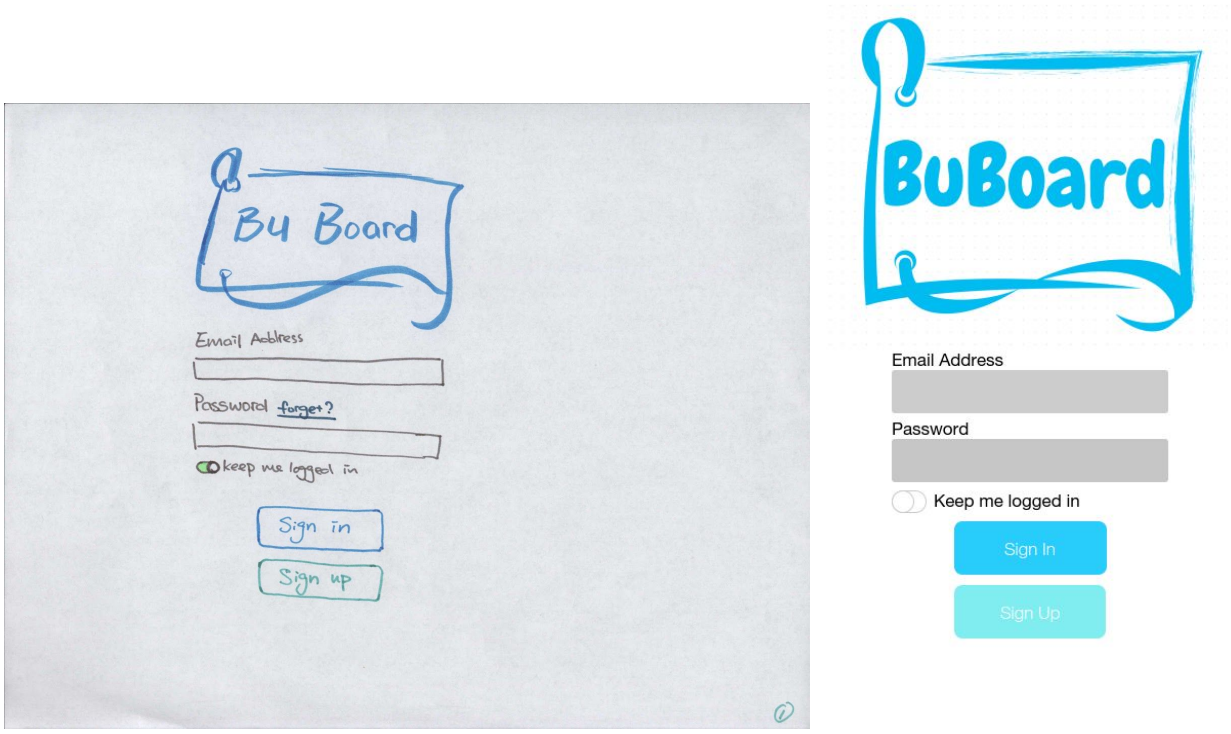



*fig 1. sign in page (left: web, right: mobile)*

This is the initial page that user will see when they open the BuBoard app. If user has an account with BuBoard already, then it requires user's email address and password to login. If user forgot his/her password, user can click the forget? link in order to set up a new password for this account. There is also a toggle switch, if user chose to switch on, then the BuBoard will automatically log in for the user in future. Otherwise, user is required to sign in everytime he/she opens BuBoard app. If user is new to BuBoard, click the 'signup' button on the bottom, it will link user to the 'create your account' page.

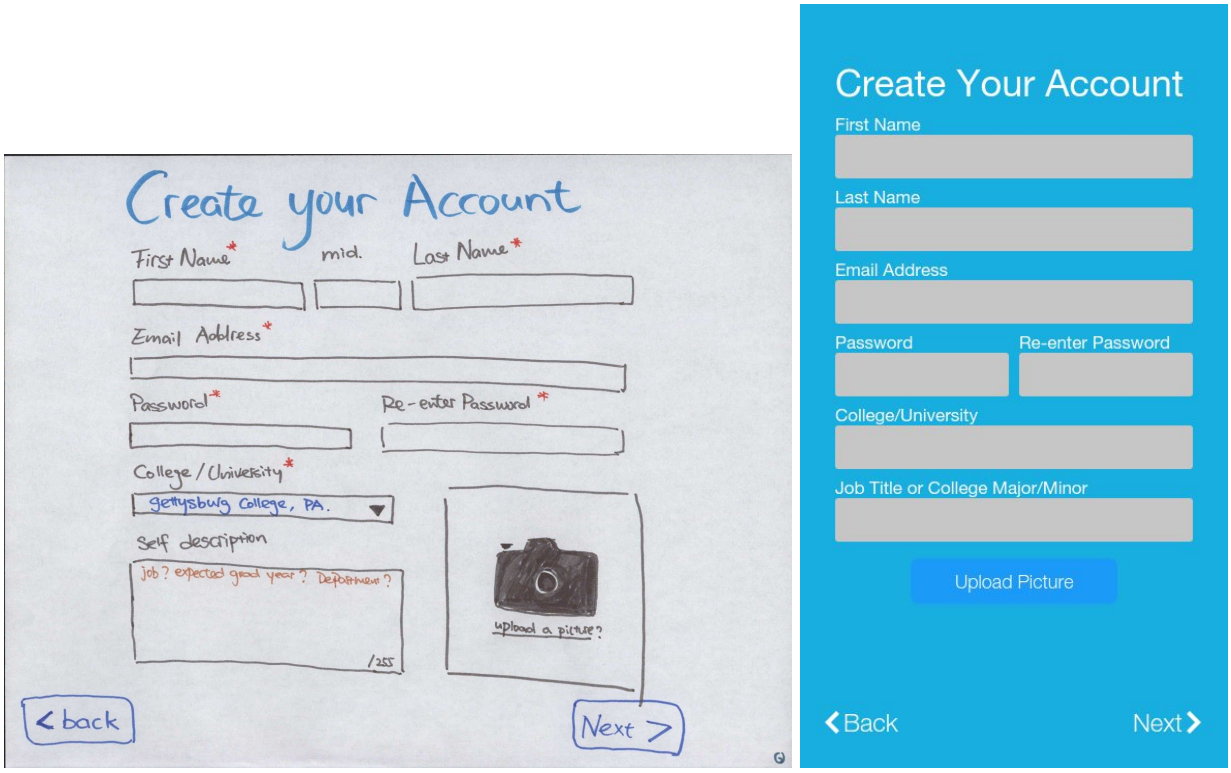



*fig 2. create your account page (left: web, right: mobile)*

This is the page when first-time user sign up their account. If there is a red star shown after the field name, then it is a required field. For example: First name, Last name, Email address, Password, Re-enter password and college are all required fields. If user tries to hit next button without fully completing this form, a pop-up box will appear, saying that there are missing fields. In the self description box, user can specify whether he/she is a student or a faculty, which department is he/she in, etc. Fields ends with no red star means they are optional fields, which means user has the right to hide his/her identity. If for some reason user wants to quit during the registration process, by clicking the back button BuBoard will take user back to the login page, as fig.1 . Once user has completed this form, click the next button, it will lead user to the verification page.

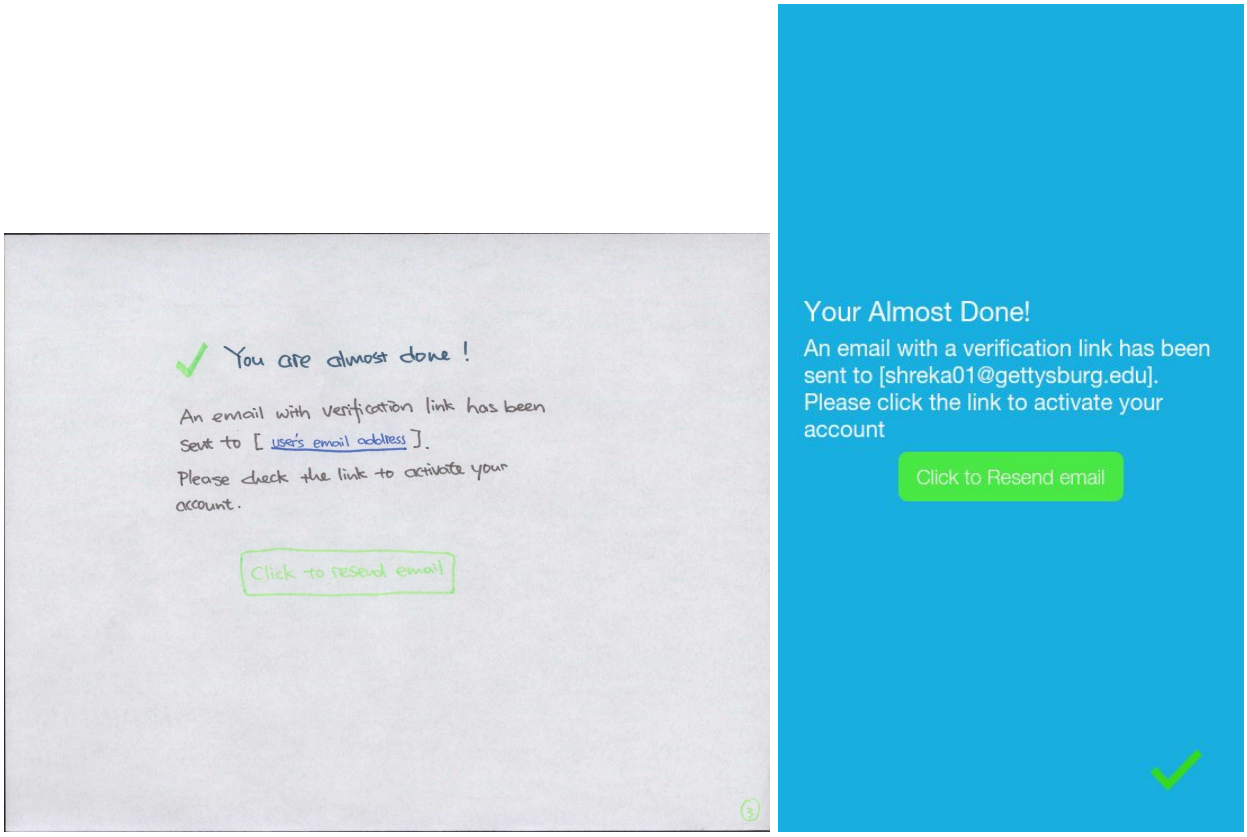



*fig 3. verification page (left: web, right: mobile)*

This page will only appear when user has successfully completed the registration process. BuBoard system will send a verification email to the email address which is associated with user newly created account. After clicking the link in that email, the full sign up process has been completed. If for some reason user didn't receive the verification email, then please click the resend email button to get a new confirmation email.

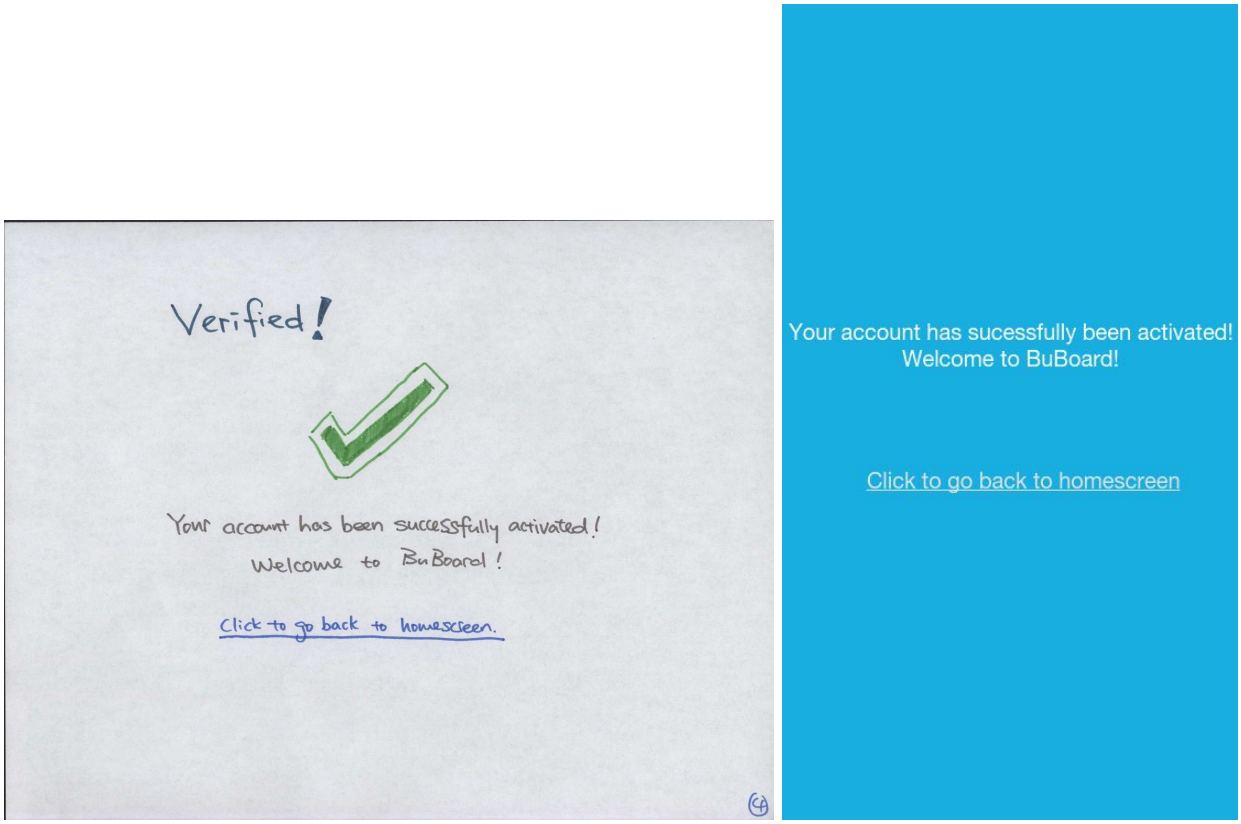



*fig 4. verified page (left: web, right: mobile)*

If user received the verification email and has clicked the verification link, our system will noticed. The verified page will show and user can go to BuBoard home page by clicking the ’go back to homescreen’ link.

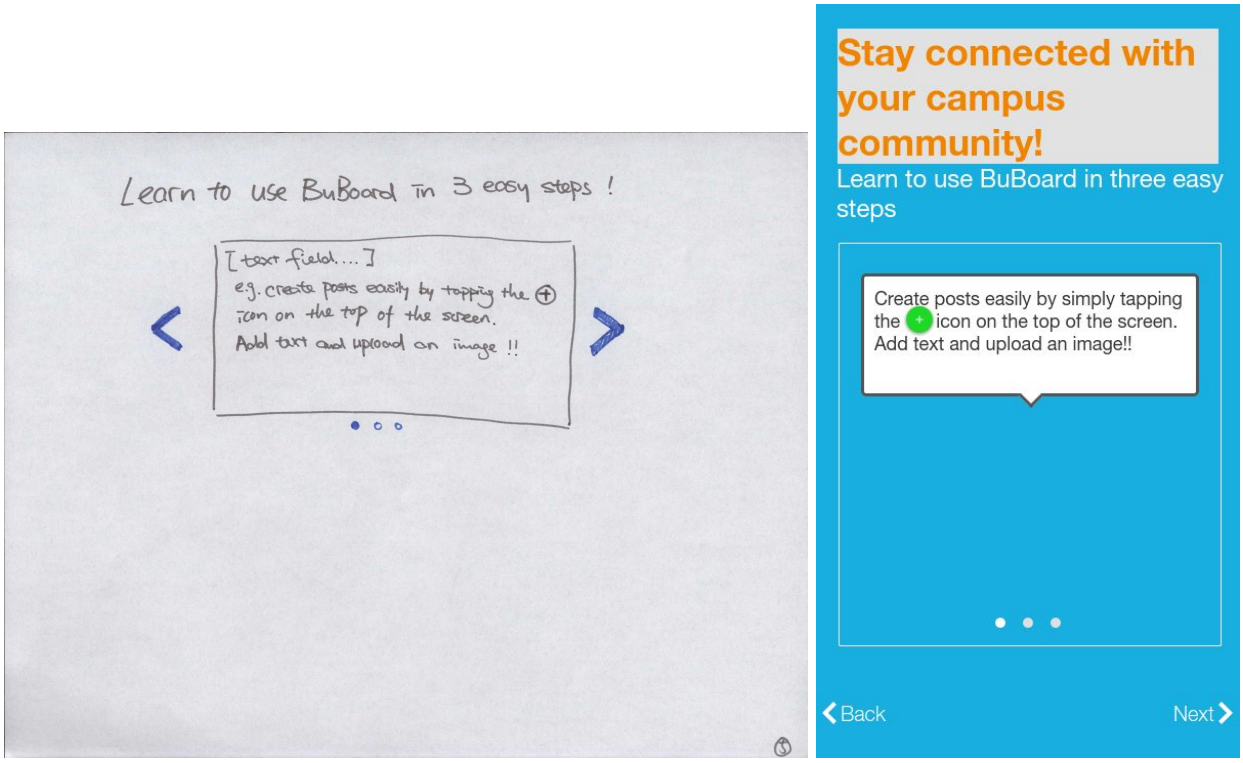



*fig 5. help page (left: web, right: mobile)*

This tutorial will automatically pop-up for the first time user and it cannot be skipped. Detailed instruction is included. If during the later stage user still wants to look the help tutorial, it can be found from the side navigation bar.

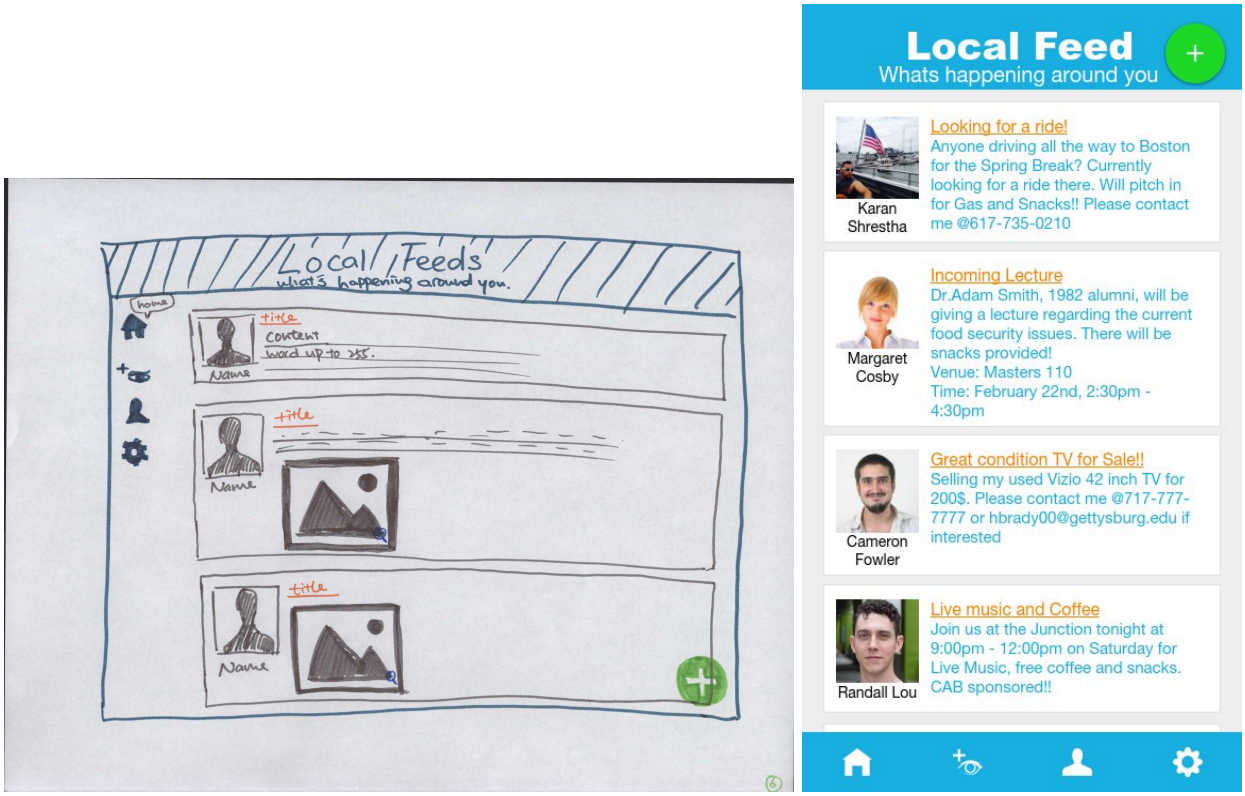



*fig 6. home page (left: web, right: mobile)*

Welcome to BuBoard home page! Our homepage consists of a header (with shaded background, saying 'local feeds'); a side navigation bar (shown on the left side) which has 4 distinct icons; and the posts. Each icon links to different web pages, and the side navigation bar exists on each BuBoard page. The first one is the home button, no matter which page user is currently at, it will always take user back to the home page, which is the 'local feed'. It may also contain a sub list of other pages. e.g. Personal feeds, miscellaneous, etc. (Haven't decided the final solution for this). The second one allows user to follow other accounts/ add friends by entering users' Gettysburg email addresses. The third one is the personal profile page, where user can reset their password, changing profile picture, etc. The last one is the setting page, help tutorial will also be included in this link. Post wise, each post contains user's name and

profile picture, a title, some texts (optional) and/or an image, and if user clicks on the image, it will  be enlarged. The green button shown on the lower right corner is for creating a new post, the layout is shown on fig 7.

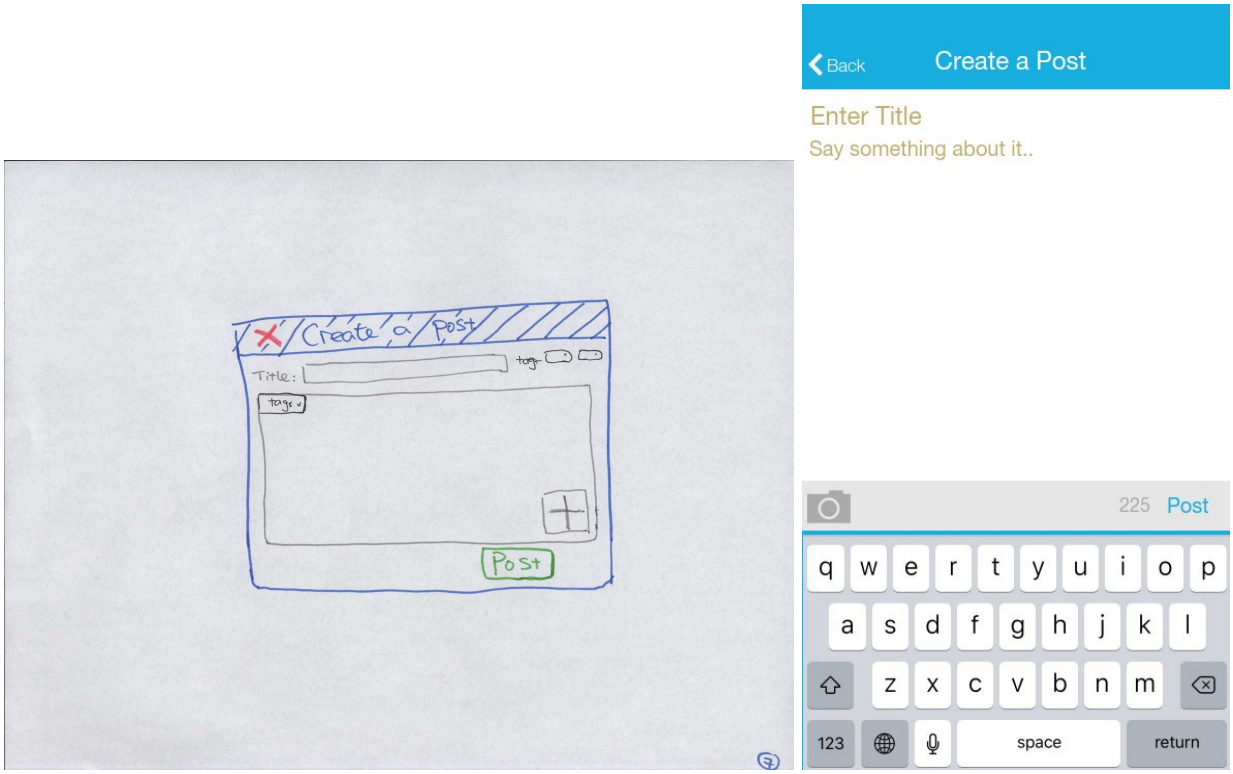



*fig 7. creating a new post page(left: web, right: mobile)*

This popup box will occur only when user clicks on the green plus button shown on fig 6. User can cancel the post by clicking the red ‘x’. Title is a required field. User can also choose tags for this post (upto 5 tags). The plus sign in the text field is for adding an image. Each post can only have one image. User post their post by clicking the post button and this pop-up box should disappear automatically.

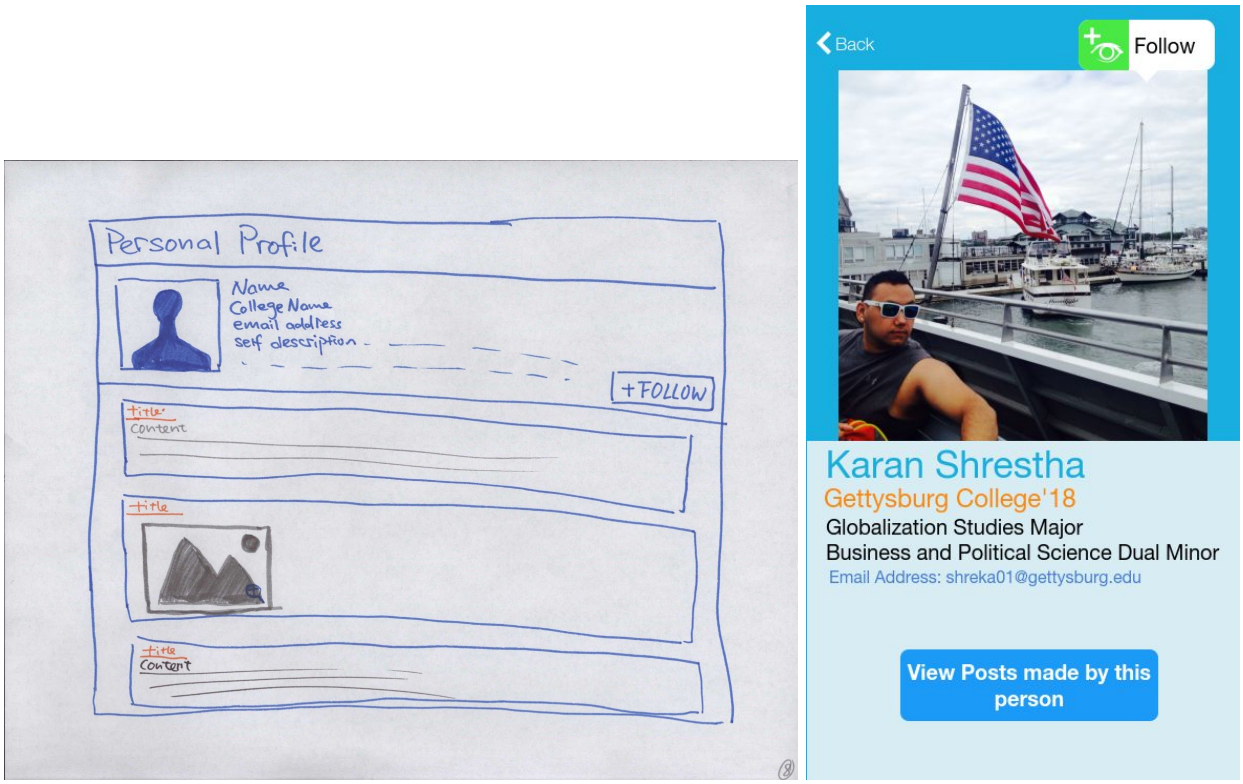
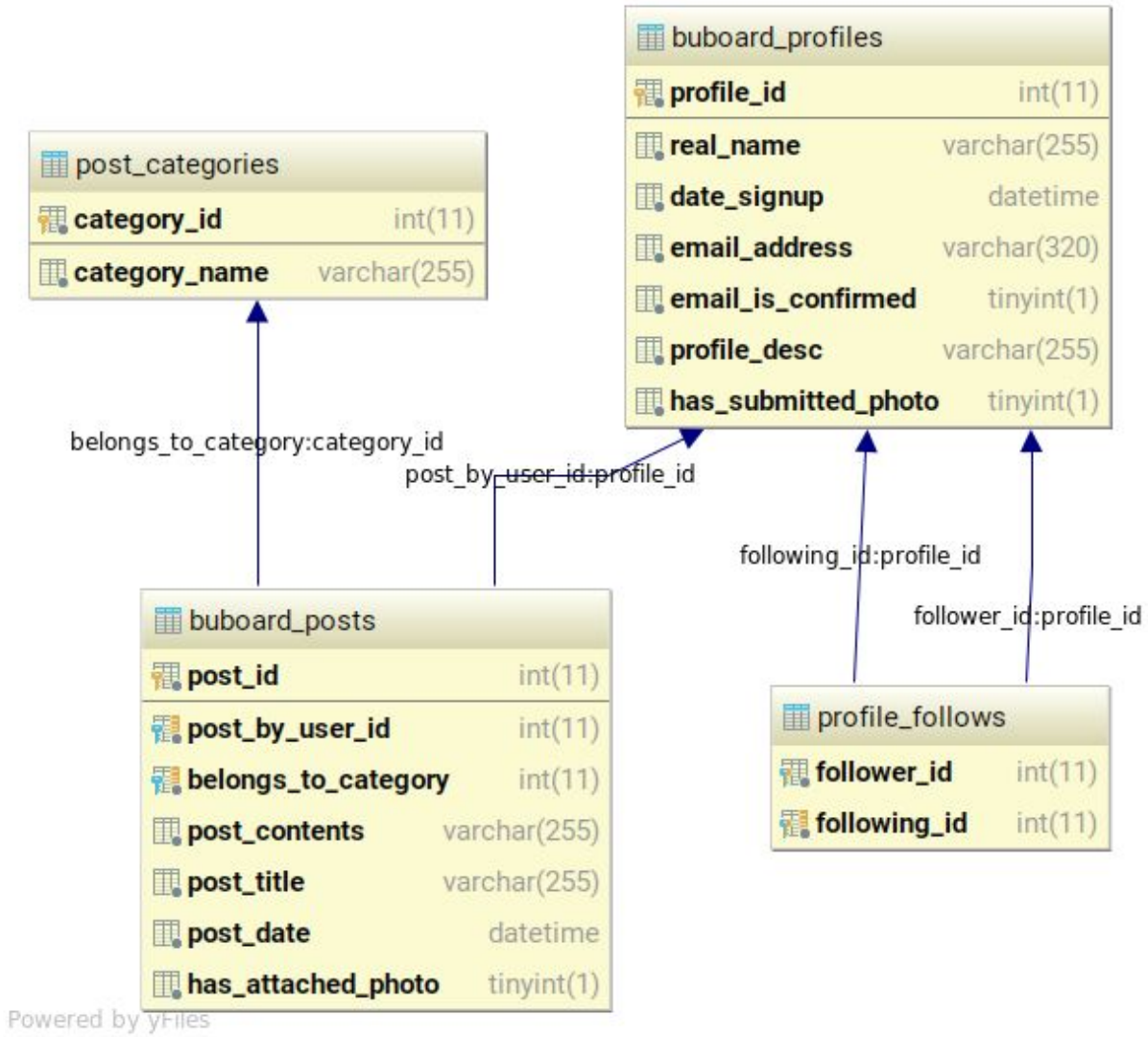



*fig 8. viewing other people's profile page (left: web, right: mobile)*

This is the standard personal profile page when user is viewing others' account. The first portion will be the general information about this user: his/her name, college, email address and self description. In case user wants to get updates on what this person posts, user can follow this person by clicking the follow button. The second portion is this person's previous posts. This part contains a scroll bar, however, the first portion of user info is fixed, or in other words, it won't be affected by scrolling the mouse.

**Models**

From the initial description of the application, it was clear that the application revolves around two different types of data objects: user's profiles, with all of their associated data, and posts, with their metadata and contents. During design, our system was created with four models,




containing a total of 19 unique fields that will store the application's data.

In the diagram above, each box represents a model. Each arrow from model to model represents a relationship, or dependency, on one model from another. The contents of the box represent the individual pieces of data that make up an entry, or row, of that model.

The first model, titled buboard_profiles, contains information for individual user-created profiles. Each profile is uniquely identified and addressed by a profile_id number, which is automatically assigned to the user upon creation. This model also stores the date and time that the user account was created, the full name of the user, a hashed copy of the user's password, used to later authenticate the user, the user's email address, a flag that lets the application know if that address had been individually confirmed, a secret key that the user would use to confirm it, a description to be displayed on the profile page, and a flag to denote if they have a profile image to display. The image is stored separately from the database.

The second model, buboard_posts, contains information relating to individual posts made to the system. They are also uniquely identified by a numeric ID. This model also contains a reference to the profile model, which allows posts to be attributed to the user that made them. A reference to another model, not yet described, attributes posts to a specific category. A field stores the text contents of the post, another stores the title, and yet another stores the date. The last field stores a flag that distinguishes if the post has a photo attached to it.

The next two models are essentially 'helpers' to the two previous structures. The third model, post_categories, is a numbered list of categories that posts may fall into. The fourth, profile_follows, contains only two fields. Both of these fields are references to buboard_profiles, and establishes the many-to-many relationship of the relation between profiles that follow each

other. When a profile A follows a profile B, an entry will be made in the form of A:B. If profile B decides to follow back, another entry B:A will be made.

**Controllers**

The controllers of an application are relatively small functions performed on request that can retrieve, transform, create, edit, or delete data from the system. The following is a list of every individual operation we have identified that this application performs.

- Authenticate User
- Check authentication status
- Register a new account
- Send registration email
- Confirm email address
- Pull and format latest posts into JSON
- Paginate latest posts response
- Filter posts by criteria
- Re-sort posts on criteria
- Pull and format a profile's follow-stream as JSON
- Follow a profile
- Unfollow a profile
- Pull and format all of a profile's posts as JSON
- Log out / End session
- Change user profile description
- Create new post category
- Create a new post
- Upload photo for new post
- Upload photo for profile
- Delete a post
- Post expiration / Prune

## Looking Forward

The long terms goals of this project, as expressed by our clients, is to make a system that is able to be easily used and implemented by colleges across the nation. Meaning that each

college would have its own individual BuBoard, which would only interact with itself and not be an inter-college system. The short term goal is to take this system to specific departments across campus and have them try the Beta versions of this product to make sure that it is as good of a product as can be designed.

**Goal List in Chronological Order**

- Complete Requirements Document
- Complete database structure
- Complete the view portion of the MVC
- Supply test web page to clients
- Fill functionality out, start adding actions
- Receive iterative feedback on web page and implement changes
- Continue iterative test process
- Finalize product and prepare for deployment
- Create public presentation

## Iterative Testing Process

Our clients have described a specific testing process that they wish to use for development of this application. This process involves taking beta versions of the application and giving it to certain departments to use for a short period of time, which will be determined by our clients, and then give feedback on what they liked and disliked about the application. Then our group would have to make these changes to the application. Once updated it would be re-released to another group and continue the process until our clients felt that this is a finalized product.

## Follow Up and Expected Client Involvement

As the application by nature is tentative and experimental, we are aware that requirements may shift and pivot as progress towards implementation makes these changes more obvious. It is our intention to inform the client of significant milestones reached as we reach them, and prepare times for demos and feedback. By our estimate, the vast majority of the functionality requested is achievable in a semester. Our clients have informed us that they have recruited an academic department to perform user testing, and the results of these tests can inform our direction in the later parts of the semester, time permitting.

## Goal Summary

The main purpose of this application is to give Gettysburg students a better way to communicate with other students across campus. This will be done through an application called BuBoard, which will function as a virtual bulletin board, allowing students to post and follow each other's comments.

## Glossary of Important Terms

1. The Digest: A Gettysburg College paper containing a large wealth of information about ongoing campus activities and other relevant information.
2. MySQL: A standardized database structure that allows for large amounts of information to be stored in a structurally simple way, and allows for quick and easy access to that information/

3. Material Design Lite (MDL): A google based programing package, which contains standardized ways to construct a web site. Its main benefit is that a web page constructed with MDL will be uniform and be look visually appealing.
4. Javascript: A HTML based programing language, which is used for creating interactivity within webpages.
5. jQuery: A library for javascript that allows web developers to add additional functionality to their web page.
6. Model: In the context of MVC, defines how an application will manage its data, logic, and internal rules.
7. View: In the context of MVC, defines what a uses will see when using this application.
8. Controller: In the context of MVC, defines the input and outputs that certain user interactions will cause.